



Mawlana Bhashani Science And Technology University

Lab-Report

Report No : 05

Course Code : ICT-3110

Course Title : Operating System Lab.

Date of Performance :

Date of Submission : 30/09/2020

Submitted By:

Name: Md. Mehedi Hasan Tipu

ID: IT-18046

3rd Year 1st Semester

Session : 2017-18

Dept. of ICT

MBSTU

Submitted To:

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU

Threads of a process on linux and thread programming

Threads: A thread is a flow of execution through the process code, with its own program counter that keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history.

A thread shares with its peer threads few information like code segment, data segment and open files. When one thread alters a code segment memory item, all other threads see that.

A thread is also called a **lightweight process**. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

Kinds of Thread:

Threads are implemented in following two ways –

- **User Level Threads** – User managed threads.
- **Kernel Level Threads** – Operating System managed threads acting on kernel, an operating system core.

Multithreading Models:

Some operating system provide a combined user level thread and Kernel level thread facility. Solaris is a good example of this combined approach. In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the entire process. Multithreading models are three types

- Many to many relationship.
- Many to one relationship.
- One to one relationship.

Corresponding Code:

```
#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>

pthread_t tid[2];

void* doSomething(void *arg)
{
    unsigned long i = 0;
    pthread_t id = pthread_self();

    if(pthread_equal(id,tid[0]))
    {
        printf("\n First thread processing\n");
    }
    else
    {

```

```

    printf("\n Second thread processing\n");
}

for(i=0; i<(0xFFFFFFFF);i++);

return NULL;
}

int main(void)
{
    int i = 0;
    int err;

    while(i < 2)
    {
        err = pthread_create(&(tid[i]), NULL, &doSomething, NULL);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
        else
            printf("\n Thread created successfully\n");

        i++;
    }

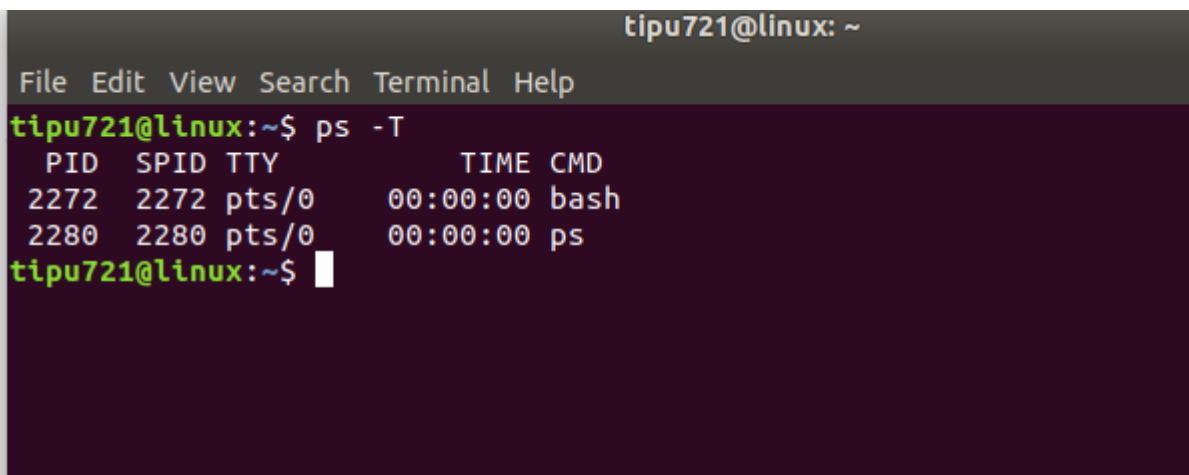
    sleep(5);
    return 0;
}

```

Thread in command line:

Here are several ways to show threads for a process on Linux.

1: PS*In `ps` command, "-T" option enables thread views. The following command list all threads created by a process with `<pid>`*



```

tipu721@linux: ~
File Edit View Search Terminal Help
tipu721@linux:~$ ps -T
  PID  SPID  TTY      TIME  CMD
  2272  2272  pts/0    00:00:00  bash
  2280  2280  pts/0    00:00:00  ps
tipu721@linux:~$

```

The "SID" column represents thread IDs, and "CMD" column shows thread names.

2: Top:

The `top` command can show a real-time view of individual threads. To enable thread views in the `top` output, invoke `top` with "-H" option. This will list all Linux threads.

```
tipu721@linux: ~
File Edit View Search Terminal Help
tipu721@linux:~$ top -H

top - 10:31:02 up 18 min,  1 user,  load average: 0.41, 0.34, 0.36
Threads: 569 total,   4 running, 529 sleeping,   0 stopped,   0 zombie
%Cpu(s): 31.8 us,  3.2 sy,  0.0 ni, 64.6 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 2038676 total,   75156 free, 1215012 used,  748508 buff/cache
KiB Swap: 728520 total,  728520 free,    0 used.  594808 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1420 tipu721   20   0 3499288 301408 110212 S 27.7 14.8   1:18.53 gnome-shell
 1426 tipu721   20   0 3499288 301408 110212 R 17.7 14.8   0:34.32 llvmpipe-0
 1427 tipu721   20   0 3499288 301408 110212 R 16.8 14.8   0:34.05 llvmpipe-1
 1227 tipu721   20   0 704308 229340 101988 R  4.2 11.2   0:40.50 Xorg
 1234 tipu721   20   0 704308 229340 101988 S  2.3 11.2   0:06.72 InputThread
 2228 tipu721   20   0  49720  4448   3352 R  2.3  0.2   0:00.72 top
    1 root       20   0 159936  9276   6776 S  0.3  0.5   0:02.94 systemd
   595 root       20   0  38756  3332   3048 S  0.3  0.2   0:00.01 cron
 1369 tipu721   20   0 258356  2916   2536 S  0.3  0.1   0:01.73 dndX11
 1398 tipu721   20   0  50052  4420   3820 S  0.3  0.2   0:00.24 dbus-daemon
 1437 tipu721   20   0 3499288 301408 110212 S  0.3 14.8   0:00.08 JS Helper
 1438 tipu721   20   0 3499288 301408 110212 S  0.3 14.8   0:00.08 JS Helper
 1568 tipu721   20   0 1090164 23768  18536 S  0.3  1.2   0:00.51 gsd-media-k+
```

Discussion:

The operating system tracks processes through a five-digit ID number known as the pid or the process ID. Each process in the system has a unique pid. Pids eventually repeat because all the possible numbers are used up and the next pid rolls or starts over.